

# FLEXIBLE ADAPTATION ENGINE FOR ADAPTIVE TRANSVERSAL FILTERS

## Priority Claim

[0001] This application claims the benefit of U.S. provisional application no. 60/461,065, filed April 7, 2003, the disclosure of which is hereby incorporated by reference.

## Background of the Invention

### Field of the Invention

[0002] The present invention relates to communications systems that use digital filters. More specifically, the invention relates to adaptation circuits and methods for adjusting the filter coefficients of a transversal or other digital filter.

### Description of the Related Art

[0003] Filtering is a common and powerful function that finds use in a large variety of applications. One important application is communications, in which information is sent from one place to another over a channel. A filter used to compensate for the effects of the channel is commonly referred to as an equalizer.

[0004] A major source of error in information transmission is inter-symbol interference (ISI), which arises when a signal is sent across a dispersive channel. Dispersive channels tend to spread the energy of a transmitted signal out over time, which means both past and future symbols can interfere with the current symbol.

[0005] To further illustrate this point, consider a transmitted signal,  $x[k]$ , which is sent across a dispersive channel with impulse response  $h[k]$ . The received signal,  $y[k]$ , is given by:

$$\begin{aligned}
 y[k] &= \sum_n h[n]x[k-n] \\
 &= h[0]x[k] + \sum_{n<0} h[n]x[k-n] + \sum_{n>0} h[n]x[k-n]
 \end{aligned}
 \tag{Eqn 1}$$

[0006] The second term in equation 1 arises from the precursor component of the channel impulse response, and allows future symbols to interfere with the current symbol.

The third term in equation 1 arises from the postcursor component of the channel impulse response, and allows previous symbols to interfere with the current symbol. Fortunately, equalization can be used to reduce or remove these components.

[0007] Oftentimes, one has no prior knowledge of the channel characteristics, making it difficult to define the optimum filter. To overcome this problem, filters are often made adaptive, allowing them to “learn” the channel characteristics.

### Adaptive Transversal Filters

[0008] The adaptive transversal filter is a fundamental component in adaptive equalization applications, and is a well understood non-recursive structure. Adaptive transversal filters commonly operate in the discrete time domain and have a finite impulse response (FIR). A generalized block diagram of a typical adaptive transversal filter is shown in Figure 1. For convenience, the input history and coefficients are expressed as vectors:

$$\mathbf{X}_k = [x[k] \quad x[k-1] \quad \dots \quad x[k-N]]^T \quad \text{Eqn 2}$$

$$\mathbf{W}_k = [W_0[k] \quad W_1[k] \quad \dots \quad W_N[k]]^T \quad \text{Eqn 3}$$

[0009] Coefficient adaptation is performed based on the desired response,  $d[k]$ , and the filter output,  $y[k]$ . The desired response,  $d[k]$ , is often a training (pilot) signal, which is essentially a copy of the transmitted sequence stored in the receiver, or the hard decisions of a Decision Feedback Equalizer (DFE). Commonly used adaptation algorithms attempt to minimize the mean-square error,  $E[\epsilon_k^2]$ , where the error signal is given by:

$$\epsilon_k = d[k] - y[k] = d[k] - \mathbf{W}_k^T \mathbf{X}_k \quad \text{Eqn 4}$$

[0010] Expanding the square of the error signal gives:

$$\begin{aligned} \epsilon_k^2 &= (d[k] - \mathbf{W}_k^T \mathbf{X}_k)^2 \\ &= d[k]^2 + \mathbf{W}_k^T \mathbf{X}_k \mathbf{X}_k^T \mathbf{W}_k - 2d[k] \mathbf{X}_k^T \mathbf{W}_k \end{aligned} \quad \text{Eqn 5}$$

[0011] To produce a reasonably simplified expression for the mean-square error, the following assumptions may be made: (1)  $\mathbf{W}_k$  is fixed, and (2)  $\mathbf{X}_k$ ,  $d[k]$ , and  $\varepsilon_k$  are statistically wide-sense stationary. With these assumptions, the mean-square error reduces to:

$$E[\varepsilon_k^2] = E[d[k]^2] + \mathbf{W}^T E[\mathbf{X}_k \mathbf{X}_k^T] \mathbf{W} - 2E[d[k] \mathbf{X}_k^T] \mathbf{W} \quad \text{Eqn 6}$$

[0012] The above equation reveals that the mean-square error is a quadratic function of the coefficient vector  $\mathbf{W}$ . This quadratic function is referred to as the *error surface*, and it contains a global minimum at the optimal coefficient vector. The adaptation engine 22 attempts to “walk” the coefficients down the error surface to a point as close as possible to the optimal solution.

[0013] A variety of basic algorithms are available to converge the coefficient vector to the optimal solution, including but not limited to, Newton’s method, the steepest descent method, least-mean square (LMS), and recursive least squares (RLS). LMS is one of the most commonly used algorithms due to its ease of computation. The LMS algorithm achieves its simplicity by approximating the mean-square error,  $E[\varepsilon_k^2]$ , with  $\varepsilon_k^2$ , leading to the following coefficient update equation:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu \varepsilon_k \mathbf{X}_k \quad \text{Eqn 7}$$

[0014] In the above equation,  $\mu$  is a step-size scalar that can be used to control convergence rate and steady-state accuracy.

[0015] In the above exemplary description, the filter and associated algorithms operate on real-valued data. The extension to complex-valued data and coefficients is well known in the art and is included in the scope of the present disclosure. Similarly, in the exemplary description the optimal coefficient vector is chosen as the one that minimizes the mean square error between the filter output and the desired response.

[0016] Those skilled in the art will recognize that it may be advantageous to choose the optimal coefficient vector based on some criterion other than mean-square error. This results in forms of the coefficient updating equation that may differ significantly from equation 7. The present disclosure is not intended to limit the applicable scope of the

invention to the exemplary description and is intended to include such techniques as may be utilized by those skilled in the art.

### **Local Minima on the Error Surface**

[0017] Under certain circumstances, local minima can also exist on the error surface. Adaptation engines that become trapped on a local minimum provide a non-optimal coefficient vector, which reduces the effectiveness of the transversal filter. Local minima are typically caused by non-linear effects in the signal path, certain channel characteristics, or some combination of the two.

### **Blind Equalization**

[0018] When the desired response,  $d[k]$ , is unknown, adaptation may be performed in a blind mode. There are many algorithms capable of blindly converging an adaptive filter using higher-order statistics of the filter's input. Some prominent algorithms include Sato's algorithm and the Constant Modulus Algorithm (CMA).

### **Decision Feedback Equalizers**

[0019] An alternative to the feedforward transversal filter, known as the Decision Feedback Equalizer (DFE), was originally proposed in 1967 and showed superior performance to its linear counterpart. DFEs were later modified to be adaptive. Adaptive DFEs typically use adaptive transversal filters 20 in both feedforward and feedback roles (although the feedforward transversal filter 20 is commonly omitted), as shown in Figure 2.

[0020] The role of the feedforward section is to reduce the precursor component of the inter-symbol interference, while the feedback section reduces the post-cursor component. Traditional symbol-rate DFEs typically correct for precursor and post-cursor components spaced at integer multiples of the symbol period,  $T$ . For example, a DFE with  $N$  feedback taps can correct for post-cursor components that occur at intervals of  $T, 2T, \dots, NT$  from the current symbol.

[0021] DFEs can be implemented in analog or digital form. Digital implementations use analog-to-digital conversion circuitry to convert the filter's input signal to digital form.

[0022] DFEs are often operated in a decision-directed mode, which uses the output of a decision device as the desired signal. In this case, the error signal is given by the

difference between the decision device's output and input. This is advantageous, as it does not require a training signal to converge the adaptation engine, although convergence is inherently more difficult.

[0023] A block diagram for a decision-directed DFE is shown in Figure 3. A common error signal and adaptation engine 22 are used to adapt both feedforward and feedback sections. The generation of this error signal can be challenging, as it is generally necessary to sample and hold, and then scale, the soft decisions (input of the decision device) before subtracting them from the hard decisions (output of the decision device). This allows the delay through the decision device to be accounted for, and also prevents the hard decisions from swamping the small signal level of the soft decisions.

### **Fractionally Spaced Equalizers**

[0024] Fractionally Spaced Equalizers (FSEs) are transversal equalizers (used as a linear equalizer or the feedforward portion of a DFE) whose taps are spaced at some fraction of the symbol period. A typical choice is  $T/2$  spacing, which allows correction of both the in-phase instant and the quadrature instant in the channel impulse response.

[0025] For an ideal, jitter-free sampling clock, equalization of anything but the ideal in-phase sampling instant provides no improvement in performance. However, when a realistic, jittered clock is considered, the true sampling instant frequently varies from the ideal point. Because of this, there is an advantage to providing equalization across the entire symbol period. Thus in realistic systems, FSEs provide superior performance to symbol-rate equalizers.

### **Adaptation Engines**

[0026] Adaptation engines 22 typically update the complete set of coefficients of the adaptive traversal filter 20 continuously. This has the advantage of providing the fastest possible convergence for a given algorithm. However, the number of operations performed by the adaptation engine 22 in a given period of time is generally directly proportional to the number of coefficients,  $N$ . For a large number of coefficients, these designs are therefore commonly inefficient and impractical.

[0027] Several designs have been proposed in which the coefficients are updated sequentially and/or in groups. Examples of such designs are set forth in the following references:

- J. Sonntag, J. Stonick, J. Gorecki, et al., "An Adaptive PAM-4 5Gb/s Backplane Transceiver in 0.25um CMOS," IEEE 2002 Custom Integrated Circuits Conference, 2002, pp. (20-3-1)-(20-3-4).
- V. Wolff, R. Gooch, and J. Treichler "Specification and development of an equalizer-demodulator for wideband digital microwave radio signals," Proc. IEEE Military Communications Conference, NY, Oct. 1988.
- S. Douglas, "Adaptive Filters Employing Partial Updates," IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing., Vol.44, No.3, pp. 209-216, March 1997.
- U.S. Patent Nos. 5,517,213 and 5,157,690.

[0028] These designs generally do not provide sufficient flexibility to optimally balance computational demands, steady-state accuracy, and start-up issues. An example of a start-up issue is the failure of an LMS engine to converge to the optimal set of coefficients due to a received signal with very low Signal to Noise Ratio (SNR).

#### Summary of the Invention

[0029] The present invention comprises a flexible adaptation engine that includes a coefficient adaptation circuit that adapts the coefficients of a transversal filter, or other digital filter, using one or more adaptation algorithms, and a controller that controls the coefficient adaptation circuit. In a preferred embodiment, the controller selects the filter coefficients to be adapted, and optionally selects the adaptation algorithm(s) to be used to adapt the selected coefficients, based on one or more criteria. These criteria may include, for example, coefficient convergence status data, clock recovery status signals, bit error rate (BER) data, the current load on a processor that adapts the coefficients, information about the transversal filter's input signal, and/or manual control signals. Different numbers of filter coefficients may be adapted on different adaptation cycles.

[0030] The coefficient adaptation circuit may implement a plurality of different adaptation algorithms, in which case the controller may select between these adaptation algorithms to adapt the selected filter coefficients. In one embodiment, the architecture supports the ability to concurrently apply different adaptation algorithms to different subsets of filter coefficients.

[0031] The flexible adaptation engine may be implemented in application-specific hardware and/or as a processor that executes software. A single adaptation engine may be used on a time-shared basis to adapt the coefficients of multiple digital filters, including filters that operate in separate, independent communications channels. The invention also comprises an equalizer, such as a decision feedback equalizer or a fractionally spaced equalizer, having one or more adaptive transversal filters whose coefficients are adapted using the flexible adaptation engine.

#### Brief Description of the Drawings

- [0032] Figure 1 is a block diagram of a prior art adaptive transversal filter.
- [0033] Figure 2 is a block diagram of a prior art adaptive Decision Feedback Equalizer (DFE).
- [0034] Figure 3 is a block diagram of a prior art decision-directed DFE.
- [0035] Figure 4 is a block diagram of an adaptation engine according to one embodiment of the invention.
- [0036] Figure 5 illustrates one embodiment of the coefficient adaptation block of Figure 4.
- [0037] Figure 6 illustrates an embodiment in which the flexible adaptation engine of Figure 4 adapts the filters of a set of  $J$  transversal filters that are integrated with the adaptation engine within a single integrated circuit.
- [0038] Figure 7 illustrates an embodiment in which the transversal filters and the adaptation engine reside within separate integrated circuits.
- [0039] Figure 8 illustrates an embodiment in which the adaptation engine is implemented within software executed by a digital signal processor or a microprocessor.

[0040] Figure 9 illustrates a receiver system in which the flexible adaptation engine is used to adapt the coefficients of one or more transversal filters of an equalizer.

#### Detailed Description of the Preferred Embodiments

[0041] The present invention comprises a flexible adaptation engine design employing a coefficient selection algorithm that can balance the needs of computational demands, steady-state accuracy, and start-up issues. The ability to dynamically and flexibly select filter coefficients to be adapted allows the adaptation engine to address the time-varying nature of these separate needs. The design does not place any restrictions on the adaptation algorithm used to update the selected coefficients. In a preferred embodiment, multiple different adaptation algorithms are implemented by the flexible adaptation engine, and a selection process is used to select between these adaptation algorithms.

[0042] The flexible adaptation engine is preferably used to adapt the coefficients of a transversal filter 20. As will be recognized, the design may also be used to adapt coefficients and other parameters of other types of digital filters (including various classes of Finite Impulse Response and Infinite Impulse Response filters), and may be used to adapt parameters of other types of adaptable systems. A single flexible adaptation engine may be shared between multiple digital filters, including filters that operate in separate data transmission paths or channels, to further reduction computational demands.

[0043] Figure 4 is a top-level block diagram of a flexible adaptation engine 30 according to one embodiment of the invention. In this embodiment, a conventional adaptation engine 22 is augmented in-part with a controller block (“controller”) 32 to provide increased flexibility. As described below, the controller 32 and the Coefficient Adaptation circuit 34 may be implemented in application-specific hardware, software executed by a processor (or a cluster of processors), or a combination thereof.

[0044] The controller 32 generates a signal *csa\_sel* to select the filter coefficients to be adapted by a Coefficient Adaptation block or circuit 34. An important feature of the design is that the controller 32 can vary the number of coefficients that are selected for concurrent adaptation. For example, during one adaptation cycle, the controller 32 may select most or all of the transversal filter’s coefficients for adaptation, and on another



adaptation cycle, the controller may select a smaller subset of the filter's coefficients. Selection of a relatively small subset of coefficients may be desirable, for example, in order to reduce the load on the processor or other circuit that adapts the coefficients, and/or to reduce the time required to adapt the selected set of coefficients. As described below, one or more different coefficient selection algorithms may be implemented by the controller 32 to select the coefficients to adapt.

[0045] As illustrated in Figure 4, the controller 32 also preferably generates a signal *aa\_sel* to specify the adaptation algorithm(s) to be applied to the selected coefficients. Although many different adaptation algorithms can be used, in practice it is contemplated that the Coefficient Adaptation circuit 34 will implement only a small number (e.g., 2, 3 or 4) of adaptation algorithms. The design may alternatively be implemented such that only a single adaptation algorithm is used, or such that the Coefficient Adaptation circuit 34 selects the adaptation algorithm to use based on the number of coefficients selected. In a generalized form, any adaptation algorithm implemented within the Coefficient Adaptation circuit 34 can be applied to any coefficient of the transversal filter 20 at a given time.

[0046] In the illustrated embodiment, the controller 32 accepts a set of inputs 35 that can vary depending on the application. For example, one or more inputs may be provided for allowing a user to manually select the adaptation engine's mode of operation. The input signals to the controller 32 may also include control signals from other components, such as the Coefficient Adaptation circuit 34.

[0047] The filter output,  $y[k]$ , and desired response,  $d[k]$ , can come from any transversal filter 20, allowing a single adaptation engine 30 to operate on multiple transversal filters 20. For example, the flexible adaptation engine 30 may be used in place of the adaptation engine of Figure 3, in which case the flexible adaptation engine 30 will receive  $y[k]$  and  $d[k]$  signals from, and provide updated filter coefficients to, both transversal filters 20 of the illustrated DFE. A generalized block diagram showing how a single flexible adaptation engine 30 may service multiple transversal filters 1-J is shown in Figure 6 and is discussed below. The number of taps or coefficients  $N$  of each transversal filter 20 is typically in the range of 2-128, and more typically is in the range of 10-128.

[0048] The error calculation circuit 36 may be identical to a state of the art implementation, and therefore will not be described in detail. In blind equalization applications, the error calculation circuit 36 is not used, as the desired response,  $d[k]$ , is unknown.

[0049] Some or all of the functionality of the controller 32 and the Coefficient Adaptation circuit 34 may be implemented within software executed by a digital signal processor or microprocessor circuit. This digital signal processor or microprocessor (generally “processor”) may also execute software that is not part of the adaptation process. For example, the same processor that implements the coefficient adaptation algorithm(s) may also implement the associated transversal filter or filters 20, and possibly other types of filters. In such cases, the current load on the processor may be taken into consideration in determining how many filter coefficients are to be adapted concurrently during the next adaptation cycle, and/or in selecting the coefficient adaptation algorithm(s) to be used to adapt these coefficients.

## **Controller**

[0050] The controller 32 can be constructed in a hardware circuit and/or in the form of software that runs on a digital signal processor or a microprocessor. The controller’s primary function is to select the coefficients to be adapted, and optionally the adaptation algorithm(s) to be applied thereto, by the Coefficient Adaptation circuit 34. The controller 32 may use any one or more of the following algorithms (and/or other algorithms) to select the coefficients to be adapted: (1) a weighted selection algorithm in which some coefficients (such as those that are the most dominant and thus have the greatest impact) are adapted more often than others, (2) a sub-sampled selection algorithm in which there are times that no coefficients are being adapted, (3) a random coefficient selection algorithm, (4) a group selection algorithm in which different subsets of coefficients are adapted at a given time, (5) a sequential selection algorithm in which a single coefficient is adapted at a time, and all coefficients are rotated through in a cyclical manner (this is a specific instance of the group selection algorithm). The above algorithms can be appropriately combined with one another to produce other viable alternatives.

[0051] The controller 32 can also select from a wide variety of adaptation algorithms. The adaptation algorithms may include some or all of the following, algorithms, as well as others: (1) a least mean squares (LMS) algorithm, (2) a recursive least squares (RLS) algorithm, (3) a Constant Modulus algorithm (CMA). These are but a few of the many well known adaptation algorithms that may be used. The partial-update LMS algorithms described in the prior art overlap with a combination of the sequential coefficient selection and the LMS adaptation algorithm.

[0052] In practice, only a few different coefficient selection algorithms and adaptation algorithms will typically be used within a given implementation of the adaptation engine 30. In addition, a certain coefficient selection algorithm may be associated with a certain adaptation algorithm, as some combinations might not result in convergence.

[0053] Those skilled in the art will recognize that a variety of different input signals and criteria may be used to drive the controller 32 to achieve a variety of objectives. Some examples of possible input signals and criteria include, but are not limited to, the following: (1) manual control signals, (2) convergence status signals from the Coefficient Adaptation circuit 34, (3) clock recovery status signals from an external block, (4) signals containing information about the filter input, such as frequency and amplitude content, (5) the bit error rate (BER) in a data recovery system, (6) the steady-state performance of the transversal filter's coefficients (e.g., how much they chatter), (7) power consumption of the various circuits, (8) processor load. Any one or more of these criteria may be used to select subsets of coefficients to be adapted, and/or to select an adaptation algorithm to be used to adapt one or more selected coefficients.

[0054] By way of specific example, the controller 32 may implement a training mode to select a coefficient-selection algorithm and/or an adaptation algorithm that produces the "best" results. The following is one example of how such a training mode may be used to train a receiver that includes an adaptive transversal filter 20 and flexible adaptation engine 30 according to the invention.

1. Transmit a known, repeating pattern to the receiver and attempt to recover the data.

2. Apply coefficient-selection algorithm X and adaptation algorithm Y.
3. Measure BER of recovered data.
4. Repeat steps 1-3 using various combinations of available coefficient-selection algorithms and adaptation algorithms, each time measuring and recording the resulting BER.
5. Choose the coefficient-selection algorithm/adaptation algorithm combination that gives the best BER.

[0055] Various other criteria may be used in combination with, or in place of, BER to measure the performance of the receiver. In addition, rather than using a known, repeating pattern, CRC (cyclic redundancy code) and/or ECC (error correction code) values within the transmitted data stream may be used to determine the BER, such that training may be performed during normal operation of the receiver. Further, the above process may be modified to permit different subsets of a filter's coefficients to be adapted concurrently using different respective adaptation algorithms.

[0056] Once a particular coefficient-selection algorithm/adaptation algorithm pair has been selected for use (step #5), this pair may be used unless and until some event occurs that causes the training process (steps 1-5 above) to be repeated. Examples of such triggering events may include the following: (1) degradation of the BER below a selected threshold; (2) a switch to a different operating channel or frequency; (3) user activation of a button or switch that initiates the training process; (4) an increase in processing load (e.g., CPU utilization) above a particular threshold, (5) detection of a "battery low" signal or a "low power mode signal." In the case of (4) and (5), a less computation-intensive coefficient-selection algorithm/adaptation algorithm pair may be temporarily selected for use to reduce processing load and/or power consumption attributable to coefficient adaptation operations.

### **Coefficient Adaptation Circuit**

[0057] The Coefficient Adaptation circuit 34 can be in the form of a hardware circuit, and/or can be implemented as a processor (such as a digital signal processor or

general purpose microprocessor circuit) that executes software. A generalized block diagram is shown in Figure 5 for an N-coefficient system.

[0058] An N-shift register 38 receives the serial output signal  $y[k]$  of a transversal filter 20, and converts this signal into parallel form. The parallel version of the  $y[k]$  signal is fed to a first NxN crossbar switch 40, which allows a generalized mapping of any coefficient to any one of N Single Coefficient Adaptation (SCA) blocks or components 42, as dictated by the *csa\_sel* signal. The SCA components 42 generate up/down signals that are used to adjust the corresponding filter coefficients upward and downward from their values. The second NxN crossbar switch 44 maps the outputs of the SCA components 42 to the correct accumulators in an N-register accumulator bank 46. Both crossbar switches are controlled by the signal *csa\_sel*, which may be a multi-bit signal.

[0059] Each of the N registers in the accumulator bank 46 stores a single coefficient. The set of N coefficients currently stored in the accumulator bank 46 may represent the coefficient vector  $W$  currently in use by a corresponding transversal filter 20. Each accumulator operates generally by adjusting its respective coefficient value in response to the up and down signals received from the second NxN crossbar switch 44. If the flexible adaptation engine 30 services multiple N-tap filters 20, a separate N-register accumulator bank 46 may be provided for each such filter. Although the coefficients are stored digitally in the illustrated embodiment, they may alternatively be stored using analog memory elements.

[0060] The Single Coefficient Adaptation (SCA) components 42 are very flexible in the generalized form, allowing for selection of any adaptation algorithm. Thus, different adaptation algorithms can be applied to different coefficients concurrently. All of the operations associated with the various adaptation algorithms (multiplications, additions, averaging, etc.) are preferably performed by the SCA components 42. Common error calculation functions can be moved into the Error Calculation component 36 (Figure 4) at the top level. To enable the adaptation engine 30 to operate on subsets of coefficients, the SCA components 42 also preferably have a disable option selectable by the respective *aa\_sel* signals.

[0061] Thus, by controlling the two crossbar switches 40, 44 and selecting appropriate adaptation algorithms in the illustrated embodiment, the controller 32 (Figure 4)

can update any subset of a transversal filter's coefficients at a given time. In addition, different mutually-exclusive subsets of coefficients can be updated concurrently using different respective adaptation algorithms. For example, the controller 32 can select a first subset of one or more filter coefficients to update using a first adaptation algorithm, while selecting a second subset of one or more filter coefficients to concurrently update using a second adaptation algorithm. This may be desirable where some of the parameters being adapted have different properties or requirement than others. For example, if the first coefficient of a transversal filter 20 is the most dominant (and therefore the most important), it may be desirable to apply a more computationally intensive (and reliable) adaptation algorithm to this coefficient than to the filter's other coefficients.

[0062] In other embodiments, the controller 32 may be capable of selecting coefficients to update with a lesser degree of flexibility. For instance, the number of coefficients that may be updated at a time may be fixed, or may be limited to certain values (e.g., 2, 4, 8, ...). In addition, certain rules or restrictions may be imposed for governing which coefficients may be updated concurrently. Further, as mentioned above, the controller 32 need not be capable of selecting an adaptation algorithm to be applied.

[0063] In pure software implementations, the functionality of the crossbar switches 40, 46, the SCA components 42, the selection signals *csa\_sel* and *aa\_sel*, and the accumulator bank 46 may be implemented in software. Multiple instances of the code used to implement the SCA components 42 may be executed in parallel to permit different coefficients to be adapted concurrently, optionally based on different adaptation algorithms.

[0064] In a basic implementation, the Coefficient Adaptation circuit 34 may include one or two available adaptation algorithms and a simplified version of the mapping function. For example, in one embodiment, the single coefficient partial-update LMS algorithm described in the prior art is implemented by making the following simplifications: (1) the first crossbar switch 40 is replaced with an N:1 multiplexer, (2) only one SCA component 42 is provided, (3) the second crossbar switch 44 is replaced with a 1:N demultiplexer, and (4) only a single accumulator in the N-accumulator bank 46 (the valid output of the demultiplexer) is updated at a given time.

[0065] Figure 6 depicts an embodiment in which the flexible adaptation engine 30 services multiple transversal filters 1-J on a time-shared basis. The transversal filters 20 and the flexible adaptation engine 30 are implemented entirely in hardware on a single integrated circuit (IC) 50, which may be an FPGA, an ASIC, or another type of custom IC. In this embodiment and the other embodiments in which a single flexible adaptation engine 30 services multiple filters 20, the filters may be distributed across multiple, separate data transmission channels that are generally independent of each other; this is in contrast to the prior art DFE design shown in Figure 3, in which the two transversal filters are part of the same data transmission channel.

[0066] Another hardware only embodiment (Figure 7) shows how the various components can be placed on separate ICs, each of which can be an FPGA or a custom IC. In this embodiment, the flexible adaptation engine 30 resides within one IC 50A, and the transversal filters 20 it services reside within separate ICs 50B, 50C.

[0067] Figure 8 illustrates an embodiment in which the flexible adaptation engine 30 is implemented in software 30A executed by a processor 30B. The processor 30B is preferably either a digital signal processor (DSP) or a general purpose microprocessor. In this example, the flexible adaptation engine 30 again services multiple transversal filters 20 that are spread across multiple ICs 50D, 50E. In addition to implementing the controller and coefficient adaptation algorithms described herein, the processor 30B may implement various other algorithms associated with the transversal filters 20. For example, as a variation of the design shown in Figure 8, some or all of the transversal filters 20 may be implemented in software executed by the processor 30B.

[0068] Various other approaches are also possible. For example, the controller logic for selecting coefficients to adapt and/or selecting adaptation algorithms may be implemented in software, while the logic for adapting the selected coefficients may be implemented purely in hardware.

[0069] Figure 9 illustrates a typical configuration in which the flexible adaptation engine 30 is incorporated into a receiver 60, as might be done in a wireless, backplane, cable, or pc-board data transmission system. In this embodiment, the flexible adaptation engine adapts the coefficients of one or more transversal filters 20 of an equalizer 62. The equalizer

62, may, for example, be a decision feedback equalizer (DFE) of the type shown in Figure 2 or 3, or may be a fractionally spaced equalizer. The recovered data signal from the equalizer 62, and/or the recovered clock signal generated by a clock recovery circuit 64, may be provided directly or indirectly as control signals to the flexible adaptation engine 30, as described above, and used to select the coefficients to adapt and/or the adaptation algorithm(s) to use.

[0070] As will be recognized, the above-described embodiments may be combined with each other, and with the prior art systems described herein, to generate various additional embodiments. For example, the features of any one embodiment may be incorporated into any of the other disclosed embodiments.

[0071] It will also be recognized that the flexible adaptation engine design disclosed herein can also be used to adapt system variables other than filter coefficients. For example, the flexible adaptation engine design can also be used to adapt such parameters as receiver gain, recovered clock phase, offset correction variables, transmitter pre-emphasis, and transmitter swing. Thus, although the design is described in the context of digital filters, it is not so limited.

[0072] Although this invention has been disclosed in the context of certain preferred embodiments and examples, it will be understood by those skilled in the art that the present invention extends beyond the specifically disclosed embodiments to other alternative embodiments and uses of the invention and obvious modifications and equivalents thereof. Accordingly, the invention is defined only by the claims that follow.